

Telegram Bot: News Bot

Andrés Aguirre Juárez

Diego Reguilón Rodríguez

Adrián Román Urdiales

Director:

Carlos Gregorio Rodríguez

GRADO EN INGENIERÍA INFORMÁTICA

FACULTAD DE INFORMÁTICA

**DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y
COMPUTACIÓN**



TRABAJO FIN DE GRADO

Septiembre 2017

Resumen

Las tecnologías de la información y la comunicación han tenido una gran evolución en los últimos años. Hemos entrado en una era en la que todo el mundo está conectado a la red a través de un dispositivo.

Hoy en día, apenas quedan personas que salgan a la calle sin su smartphone, ello implica que podamos estar informados en todo momento de lo que pasa a nuestro alrededor. Basándonos en este hecho, hemos pensado en poner más fácil a la gente esa recepción de información.

Actualmente, existen numerosas aplicaciones de mensajería, las cuales, están integrando cada vez más bots conversacionales. La mayoría de la gente tiene alguna de estas aplicaciones instaladas en sus teléfonos (Whatsapp, Telegram, Messenger), por lo que nosotros hemos decidido desarrollar un bot, con un fin muy claro; mantener informada a la gente de los temas que les interesan. Para su desarrollo, hemos elegido usar Telegram como plataforma de mensajería. Telegram está creando un ecosistema propio gracias a los bots, proporcionan un servicio parecido al de una aplicación, pero sin la necesidad de descargarla.

Cuando la tecnología no había avanzado de esta forma, informarse de lo que ocurría en el mundo estaba limitado a la prensa escrita (periódicos), y años después, la televisión ofrecía otro método de información, mediante el telediario. Después apareció internet, y con ellos los diferentes periódicos pasaron a tener página Web con todas sus noticias.

Nosotros hemos dado un paso más, queremos ofrecer al usuario una manera más rápida y sencilla de mantenerlo informado de lo que ocurre en nuestro mundo. Meterse en distintas páginas web para buscar la misma información de distintos medios, en ocasiones puede llegar a ser pesado. Ante este pequeño handicap, decidimos crear un bot que enviase periódicamente las noticias de tu interés.

Por otro lado, toda la vida hemos visto en las últimas páginas de los periódicos, la programación de televisión. Hemos querido conservar esta costumbre y añadir esta funcionalidad al bot.

Como conclusión, queremos que nuestra aplicación aporte a los usuarios del bot una manera sencilla y eficaz de estar informado al instante, y ayudando así al continuo desarrollo de la tecnología de la información.

Abstract

Information and Communication Technologies have made a big improvement over the last few years. We live in a new era where not being connected to the network through any technological device, means staying behind in life.

Nowadays, few people go out the street without their smartphone, which means we are currently informed about what is going on around you. Taking this fact as a base, we have thought in facilitating people the reception of information.

There are currently many messenger services apps that are more and more often introducing conversational bots. The majority of people usually have these apps on their smartphone such as Whatsapp, Telegram or Messenger, so we have decided to develop a bot, with a clear objective: be able to maintain people informed about topics of their interest. In order to develop it, we have chosen Telegram as the messenger platform. Telegram is efficiently creating a personal ecosystem thanks to the bots, which offer a similar service to an app, but without needing to download it.

Before any technological advance, in the past, informing yourself about the latest news was limited to written press on the newspapers, and years later, television offered another information method, through the newscast. Then it was the internet that appeared, which brought the different newspapers to have a place on the network, with all its news.

Our group have gone one step ahead, offering the user an easier and quicker way to stay informed about our world. Different websites to find exactly the same information on different media, may result a nuisance. Taking advantage of this handicap, we decided to create a bot which daily periodically sent the latest news of your interest.

On the other hand, we have always read the news, and seen the newscast so we wanted to keep this habit and add this functionality to the bot.

In conclusion, our objective is that our application gives the bot users a simple and efficient way to stay informed instantly, helping in this way to the development of the information technology.

Palabras clave

- Bot
- Telegram
- Python
- Token
- RSS
- Feed
- BBDD (base de datos)

Keywords

- Bot
- Telegram
- Python
- Token
- RSS
- Feed
- Database

Índice de Contenidos

1 Introducción	1
1.1 Motivación	1
1.2 Introduction	2
1.3 Motivations	2
1.4 Objetivos	3
1.5 Estructura del trabajo	3
2 Estudio de las tecnologías.....	5
2.1 Aplicaciones de mensajería.....	5
2.1.1 Telegram.....	5
2.1.2 WhatsApp.....	6
2.1.3 Facebook Messenger.....	7
2.1.4 Conclusiones	7
2.2 Lenguajes de programación	8
2.2.1 Python	8
2.2.2 PHP.....	9
2.2.3 Conclusiones	10
2.3 Bases de datos	10
2.3.1 MySQL.....	10
2.3.2 PostgreSQL.....	11
2.3.3 Conclusiones	11
2.4 RSS	12
2.5 Herramientas de desarrollo	13
2.5.1 Google Drive.....	13
2.5.2 GIT.....	14
2.5.3 GEDIT.....	14
3 Arquitectura del sistema.....	16
3.1 Servidor de Telegram.....	16
3.2 Servidor de la universidad	16
3.3 Clientes	17
4 Descripción del sistema.....	18
4.1 Ámbito del sistema	18
4.2 Funcionalidades de la aplicación	18
4.3 Exploración de la aplicación	19
4.4 Requisitos	30
5 Puntos críticos del proyecto	31
5.1 Ley de protección de datos	31

5.2 Noticias repetidas	31
5.3 Problemas con tiempos de cómputo.....	31
5.4 Lenguaje Python	32
5.5 Botones de Telegram	32
5.6 Hilo actualizar programación	32
5.7 Entrega en junio.....	33
6 Conclusiones del proyecto.....	34
6.1 Project's conclusions	35
7 Ampliaciones futuras	37
7.1 Recomendador.....	37
7.2 Puntuación / comentarios de noticias	37
7.3 Programación de TV	38
8 Contribuciones personales	39
8.1 Andrés Aguirre Juárez.....	39
8.1.1 Conocimientos	39
8.1.2 Investigación	39
8.1.3 Pruebas previas	39
8.1.4 Desarrollo de la aplicación	40
8.1.5 Memoria	40
8.2 Diego Reguilón Rodríguez	41
8.2.1 Conocimientos	41
8.2.2 Investigación	41
8.2.3 Pruebas previas	42
8.2.4 Desarrollo de la aplicación	42
8.2.5 Memoria	43
8.3 Adrián Román Urdiales	44
8.3.1 Conocimientos	44
8.3.2 Investigación	44
8.3.3 Pruebas previas	44
8.3.4 Desarrollo de la aplicación	45
8.3.5 Memoria	45
Anexo 1: Referencias	46
Anexo 2: Bibliografía.....	47

Índice de Figuras

Figura 1. Telegram.....	5
Figura 2. WhatsApp.....	6
Figura 3. Facebook Messenger.....	7
Figura 4. Python.....	8
Figura 5. PHP	9
Figura 6. MySQL	10
Figura 7. PostgreSQL	11
Figura 8. RSS.....	12
Figura 9. GoogleDrive	13
Figura 10. Git.....	14
Figura 11. Gedit	14
Figura 12. Buscando el bot.....	19
Figura 13. Arrancar el bot	20
Figura 14. Bienvenida del bot.....	20
Figura 15. Comandos 1	21
Figura 16. Comandos 2	21
Figura 17. Comando ayuda	21
Figura 18. Canales de tv.....	22
Figura 19. Programación tv	22
Figura 20. Agregando un feed	23
Figura 21. Respuesta /addfeed	23
Figura 22. Periódicos	23
Figura 23. Temas a agregar.....	24
Figura 24. Lista de feeds.....	24
Figura 25. Botones eliminar feed	25
Figura 26. Borrado confirmado	25
Figura 27. Últimas noticias	26
Figura 28. Cancelar acción.....	26
Figura 29. Parar el bot	27
Figura 30. Envío de noticias.....	28
Figura 31. Recomendador	30
Figura 32. Recomendando feed	30

1 Introducción

En este primer capítulo, vamos a describir nuestras motivaciones a la hora de elegir este proyecto como trabajo de fin de grado, cómo nos podemos beneficiar de sus servicios en la actualidad y cómo se está focalizando el mundo de la tecnología en este sector.

La tecnología ha evolucionado de tal manera, que como anunció Luis Miguel Gil Pérez (presidente de Telefónica España), “en el año 2016, el Smartphone se coronó como el principal dispositivo a través del cual los españoles entran en Internet, superando por primera vez al ordenador”. Esto, es debido en gran medida, al uso de las redes sociales y aplicaciones de mensajería como medio de comunicación. Las tecnologías se adaptan a los usuarios, y ésta en concreto, cada vez integra más servicios.

1.1 Motivación

Vivimos una revolución digital única en la historia, donde la conectividad ha pasado de ser un fin a ser un medio para infinidad de cosas.

Hoy en día, hay millones de aplicaciones diferentes, con acceso a todas ellas en tu bolsillo. Los móviles nos permiten tener conectividad 24 horas, por lo que hemos querido desarrollar una aplicación para ellos. Al tenerla al alcance de la mano, su uso se disparará considerablemente. En concreto, nos hemos centrado en aplicaciones de mensajería. La mayoría de los usuarios, las usan para comunicarse con otras personas, enviar archivos, juegos, información, etc.

Hemos elegido desarrollar una aplicación sobre temas de información periodística, que se adapte a esta era digital. Normalmente, solíamos comprar el periódico para informarnos de lo que había pasado el día anterior, ahora además de ello, tenemos la oportunidad de verlo a través de las webs, de forma online. Las webs nos ofrecen una ventaja frente a los periódicos; tienen la información constantemente actualizada y no tienes la necesidad de esperar al día siguiente para ver la noticia que ha ocurrido el día anterior.

En este sentido, hemos pensado juntar las ventajas que te ofrecen las webs de diferentes periódicos, con las aplicaciones de mensajería. En concreto la aplicación de Telegram.

Decidimos integrar nuestra idea en un bot para Telegram, ya que te permite encontrar la información deseada sin salir de la aplicación.

El bot lo desarrollamos como un lector de RSS, ya que es la forma más sencilla de filtrar las noticias por temática.

Por otro lado, decidimos no perder una de las últimas páginas de los periódicos, la programación de televisión, de esta forma el usuario puede consultar de una forma rápida y cómoda qué ver en la televisión.

1.2 Introduction

In this first chapter, we will describe our motivations when choosing this project as an end-of-degree project, how we can benefit from its services nowadays and how the world of technology in this sector is being focused.

Technology has developed in a way that, since 2016, the Smartphone has been the main device through which Spanish people enter into the Internet, overpassing for the first time the computer.

This is because a large extent use of social networks and messaging applications as a way of communication. The technologies adapt to the users, and this in particular, every time fits in more services.

1.3 Motivations

We live a unique digital revolution in history, where connectivity has gone from being an end to being a means to lots things.

Nowadays, there are millions of different applications, with access to all of them in your pants pocket. The mobiles allow us to have connectivity 24 hours, and for that, we have focused on this technology. By having it within reach, its use will increase considerably. In particular, we have focused on messaging applications. Most users use them to communicate with other people, send files, games, information, etc.

We have chosen to develop an application about topics of journalistic information, which is adapted to this digital era. Usually, we used to buy the newspaper to inform us of what had happened the previous day, now besides that, we have the opportunity to see it through the webs, online. The websites offer us an advantage over newspapers; they have the information constantly updated and you do not have to wait until the next day to see the news that happened the previous day.

In this sense, we have thought to combine the advantages offered by the websites of different newspapers, with the applications of messaging. Specifically the Telegram application. We decided to fit in our idea into a bot for Telegram, as it allows you to find the desired information without leaving the application.

The bot we developed as an RSS reader, is the easiest way to filter the news of your interest, according to the desired theme. On the other hand, we decided not to lose one of the last pages of newspapers, the newcast, this way the user can quickly and confort consult what to watch on television.

1.4 Objetivos

Los principales objetivos planteados en este proyecto, son los siguientes:

- Integrar un lector de RSS en una plataforma de mensajería. De esta forma podemos leer los RSS que el usuario desea.
- Aplicar técnicas web scraping y procesamiento de la información estructurada ara unificar servicios y ofrecerlos de manera cómoda a los usuarios de Telegram
- Familiarizarse con la API de Python para Telegram para el manejo cliente servidor.
- Manejo de base de datos MySQL para poder guardar información.
- Desarrollo de código modular, para facilitar ampliaciones futuras.
- Desarrollar un recomendado para incluir componente social al bot.

1.5 Estructura del trabajo

La memoria de este Trabajo de Fin de Grado consta de la siguiente estructura:

El **Capítulo 1** contiene una breve introducción, además de la motivación que nos ha llevado a realizar este proyecto a lo largo del año, los objetivos y la estructura del Trabajo de Fin de grado.

El **Capítulo 2** describe y valora las tecnologías que se pueden emplear en el proyecto desarrollado. También se explica porque se emplea la aplicación de mensajería Telegram y no otras, se analiza cada componente de la aplicación y se decide cuál de ellos es el más adecuado.

El **Capítulo 3** describe toda la arquitectura del sistema, definiendo los componentes que la forman y explicando cada una de ellas.

El **Capítulo 4** describe el funcionamiento del sistema, sus aplicaciones y requisitos como cliente y servidor. Además, se demuestra la funcionalidad del bot desarrollado.

El **Capítulo 5** describe todos los problemas y dificultades que se han tenido a lo largo del proyecto. Además, se explica cómo se ha conseguido solucionar estos problemas y el tiempo que se ha tardado en arreglarlos.

El **Capítulo 6** describe las conclusiones obtenidas al final del proyecto.

El **Capítulo 7** describe las ampliaciones futuras del proyecto.

El **Capítulo 8** describe las contribuciones personales de cada miembro del grupo, los puntos de vista y las sensaciones que se han tenido a lo largo del Trabajo de Fin de Grado.

Por último, en el **Anexo A** se encuentran las referencias utilizadas para el desarrollo de esta memoria, y en el **Anexo B** se recoge la bibliografía, información que se ha obtenido para poder explicar los diferentes componentes utilizados en el proyecto y gracias a ello poder desarrollar el código de la aplicación.

2 Estudio de las tecnologías

En este capítulo se van a comentar los distintos estudios que se han realizado de las tecnologías disponibles y cuáles son los más apropiados para el proyecto desarrollado.

2.1 Aplicaciones de mensajería

Como ya se ha comentado, este tipo de aplicaciones son las más utilizadas y descargadas por los usuarios en la actualidad. Antes de llevar a cabo el proyecto, se realizó un estudio sobre las aplicaciones de mensajería más descargadas. Se investigó cuál de todas era la más apropiada para el proyecto, concretando su funcionamiento y averiguando si tenían soporte para desarrollar un bot.

2.1.1 Telegram



Figura 1. Telegram

Telegram Messenger es una aplicación de mensajería instantánea gratuita, privada y accesible, que fue desarrollada en el año 2013 pero hasta el 2014 no tuvo una acogida mediática. Desde aquel momento es una de las aplicaciones con mayor número de descargas.

Se puede utilizar en teléfonos, tabletas y ordenadores. Además, se encuentra disponible para más de 10 sistemas operativos: Android, IOS, macOS, Windows, GNU/Linux o navegadores web.

Telegram está logrando hacerse su propio hueco gracias a que cada vez se diferencia más de su principal competidor, WhatsApp. Una de las diferencias principales es que Telegram utiliza la arquitectura propia MTProto, con almacenamiento en la nube para los mensajes, bot, canales y supergrupos.

El usuario es el que decide si quiere descargar las fotos, vídeos y archivos en el dispositivo para ahorrar memoria o almacenarlos en la nube.

Otra característica que le diferencia de WhatsApp es que al ser de código abierto, los desarrolladores pueden crear sus propias aplicaciones dentro de **Telegram**.

Otro tipo de servicio que ofrece es la mensajería masiva en grupos y en los canales. Se puede llegar a tener un supergrupo con más de 10000 integrantes, los cuales todos los usuarios pueden interactuar entre ellos mandando mensajes de vídeo, animaciones gif, sticker. Los canales son espacios de difusión unilateral para compartir contenido a una cantidad ilimitada de usuarios.

Uno de los puntos fuertes de esta aplicación que lo diferencia del resto es el poder desarrollar un bot: es un programa con el que se puede interactuar dentro de la aplicación **Telegram** como si fuese una persona. Existen los bots que emulan conversaciones normales y los bots interactivos. En estos últimos se pueden tener conversaciones mediante botones o comandos.

2.1.2 WhatsApp



Figura 2. WhatsApp

WhatsApp es la aplicación de mensajería más utilizada en todo el mundo. Millones de usuarios pueden comunicarse gracias a su fácil acceso y su efectividad en la entrega de mensajes.

El funcionamiento de WhatsApp es bastante sencillo ya que lo único que se tiene que hacer es instalar la aplicación en el teléfono. Dicha aplicación usa el número de teléfono como código de usuario. Así, la aplicación se sincroniza con la agenda telefónica del usuario y todos aquellos usuarios que utilicen WhatsApp aparecerán

automáticamente en la aplicación. Ésta es una gran ventaja frente a las otras aplicaciones de mensajería ya que no se tienen que agregar usuarios de manera individual.

La seguridad es otra de las prioridades de WhatsApp, y esto se ve reflejado desde que se comienza a hablar a un nuevo contacto, momento en el cual la aplicación recuerda al usuario que gracias a un sistema de cifrado las conversaciones estarán protegidas de extremo a extremo.

2.1.3 Facebook Messenger



Figura 3. Facebook Messenger

Facebook Messenger es una aplicación de mensajería que fue lanzada por la red social Facebook, disponible para Android, iOS y Windows Phone. Su gran ventaja es que permite chatear con sus contactos de Facebook.

Esta aplicación también ha decidido apostar por la integración de bot. Además, los bots actúan gracias a la inteligencia artificial de Facebook. Los bots de Facebook Messenger sirven para ayudar a los usuarios a realizar compras en línea, hacer reservas, jugar a juegos online, consultar información.

2.1.4 Conclusiones

La elección de la plataforma entre los componentes del grupo fue unánime. Se escogió Telegram por el hecho de poder desarrollar un bot dentro de una aplicación de código abierto y además porque estaba formado por una gran comunidad de desarrolladores. Al tener dicha comunidad, una gran parte de los problemas que encontramos estaban solucionados, y en caso de no estarlo, el tiempo de respuesta era notablemente rápido.

2.2 Lenguajes de programación

En este apartado se realiza un estudio de qué lenguaje de programación es el más apropiado para desarrollar nuestra aplicación dentro de la plataforma Telegram. Además, se explica cómo se puede desarrollar el bot a partir de los lenguajes propuestos.

2.2.1 Python



Figura 4. Python

Python es un lenguaje de programación gratuito, multiplataforma, de código abierto y fácil de utilizar. Python fue creado por Guido van Rossum a finales de los 80 y en 1991 publicó la primera versión de código 0.9.0. Desde aquel entonces se aceptó como un lenguaje de secuencia de comandos de elección para el geoprocesamiento de usuarios y continúa en crecimiento hasta el día de hoy, con la última versión Python 3.6.2, siendo uno de los lenguajes de programación más empleados para el desarrollo software.

Este lenguaje es fácil de aprender y excelente para principiantes e incluso para expertos. Además, es altamente escalable para grandes proyectos o para pequeños programas. Puede ser utilizado en diversas plataformas y sistemas operativos, como Windows, Mac OS X y Linux, pero también puede funcionar para teléfonos, como por ejemplo Nokia, el cual desarrolló un intérprete para su sistema operativo Symbian.

Empresas y organizaciones bastante conocidas hacen uso intensivo de este lenguaje para desarrollar sus productos y servicios como es el caso de Light & Magic, NASA, Google y Nokia.

Para desarrollar un bot en la plataforma de Telegram, el lenguaje de programación que más se utiliza es Python. Con este lenguaje podemos desarrollar cualquier tipo de software para comunicaciones en red, para aplicaciones de escritorio, para crear juegos y sobre todo para aplicaciones web.

Gracias a este lenguaje de programación se pueden emplear sus propias librerías para desarrollar el bot, además de conseguir un código mucho más limpio.

2.2.2 PHP



Figura 5. PHP

PHP es un lenguaje de programación de código abierto, es utilizado especialmente para el desarrollo web y puede ser incrustado en HTML. Apareció en el año 1995 y fue diseñado por Rasmus Lerdorf pero la implementación principal de PHP es producida por el The PHP Group. Con el paso de los años ha ido mejorando hasta llegar a la última versión más estable: 7.1.8. Este lenguaje forma parte del software libre publicado bajo la licencia de PHP.

Lo que distingue a PHP es que el código es ejecutado en el servidor, generando un HTML y enviándolo al cliente. Éste recibirá el resultado tras ejecutar el script. El servidor web también puede ser configurado para que procese todos los ficheros HTML con php, de tal forma que los usuarios no pueden saber que lleva por debajo.

Al igual que el lenguaje de programación Python, se pueden desarrollar los llamados bots de Telegram sin problema ya que con PHP se pueden hacer llamadas a una dirección HTTP. También se puede utilizar alguna de las librerías de PHP para Telegram Bot de código libre.

2.2.3 Conclusiones

La decisión de utilizar un lenguaje de programación fue gracias a nuestro tutor, el cual nos recomendó utilizar Python ya que es fácil de utilizar para aquellas personas que estén empezando. También nos sugirió utilizar la librería Telepot. Además, la mayoría de bots que se han visto están desarrollados en este lenguaje y se encontró mayor documentación que en PHP. Gracias a todo este planteamiento se decidió elegir el lenguaje de programación Python.

2.3 Bases de datos

En este apartado se realiza un estudio de qué base de datos relacional es la más apropiada para el proyecto desarrollado.

2.3.1 MySQL



Figura 6. MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos. El lanzamiento fue en 1995. Es considerada como la base de datos open source más popular del mundo.

Fue escrito en C y C++ y destaca su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación como Python o PHP. Además, se puede integrar en los distintos sistemas operativos.

Es destacable que su utilización sea gratuita e incluso que se pueda modificar con total libertad, pudiendo descargar su código fuente.

La principal ventaja que tiene es su gran velocidad al conectar con el servidor como al servir selects y demás. Otra característica es que consume muy pocos recursos, tanto de CPU como de memoria. Además, aunque se cuelgue, no suele perder información y no corrompe los datos.

2.3.2 PostgreSQL

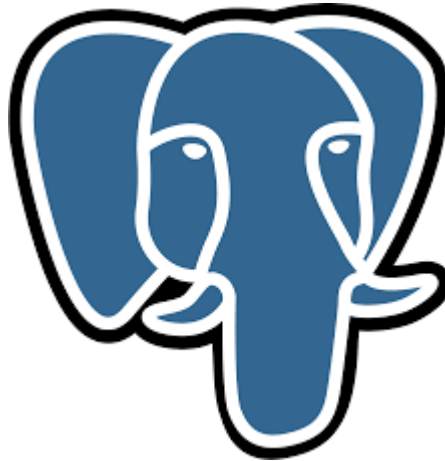


Figura 7. PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional y libre, el cual agiliza la interacción de cliente, servidor y base de datos. Su desarrollo comenzó hace más de 16 años, hasta la última serie de producción que es la 9.1. Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado.

Además, nos proporciona diferentes características como potencia, robustez. Estas características son las que más se han tenido en cuenta durante su desarrollo.

PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

2.3.3 Conclusiones

Al estudiar estos dos sistemas de gestión de base de datos relacional se decide por MySQL. La principal razón es que en el proyecto se necesitaba rapidez para hacer consultas en la BBDD y PostgreSQL es 2 o 3 veces más lenta. Además, se observa el consumo de recursos entre los dos y se vio que PostgreSQL consume bastantes recursos y carga mucho en el sistema.

También era importante ver si se podían perder datos y si se quedaba colgado el servidor, y, como se ya ha comentado anteriormente, es una de las principales características de MySQL.

2.4 RSS



Figura 8. RSS

RSS es una pieza fundamental en el proyecto desarrollado. Está compuesto por un formato XML para distribuir el contenido. Se utiliza para compartir información actualizada de una fuente de noticias. Para poder recibir información de la fuente de noticias, ésta debe tener disponible el servicio RSS y tener un lector RSS.

La principal ventaja que ofrece los sistemas RSS es que el usuario va a ser el encargado de elegir las fuentes de información de las que se quiere tener información actualizada. Otra característica es que los RSS están libres de Spam. Además, el uso de los RSS es gratuito.

En el proyecto se crea un lector RSS dentro de la plataforma Telegram. Éste es el encargado de mantener informado al usuario de todas las fuentes a las que esté suscrito y enviar los contenidos actualizados, ahorrando tiempo al usuario y no teniendo que visitar todas las páginas de interés.

2.5 Herramientas de desarrollo

En este apartado se van a comentar todas las herramientas de desarrollo utilizadas en el proyecto.

2.5.1 Google Drive



Figura 9. GoogleDrive

Google Drive es un servicio que permite a los usuarios almacenar de forma centralizada todos sus archivos de Google Docs y sincronizar estos archivos en todos sus dispositivos. Nos proporciona espacio único para almacenar, crear, modificar, compartir y acceder a documentos , archivos de todo tipo.

Lo bueno que tiene Google drive es que todos los archivos que se han subido pueden ser accesibles desde el navegador web o desde la aplicación del móvil.

Además, el componente Google Docs nos permite tratar documentos online alojados en la nube y que ofrecen la colaboración en tiempo real. Por tanto, nos pareció perfecto, la utilización de drive para realizar la estructura de la memoria y pudiéramos estar al tanto de las modificaciones que se hacían, para así estar en continuo contacto desde cualquier lugar.

2.5.2 GIT



Figura 10. Git

Git es un software de control de versiones. Git fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando existe un gran número de archivos de código fuente.

Utilizar Git en nuestro proyecto fue lo ideal, porque nos ayudó a desarrollar un trabajo en equipo como ha sido este proyecto y además de forma rápida y de manera inteligente.

Decidimos utilizar esta herramienta por llevar un control de versiones de la aplicación. Nos facilitaba el desarrollo del código sin tener que reunirnos físicamente.

2.5.3 GEDIT



Figura 11. Gedit

Gedit es un editor de código en donde se puede procesar textos libre, viene instalado en nuestros equipos con Ubuntu, o más bien todas las distribuciones que tiene Gnome. Se caracteriza porque es fácil de utilizar gracias a su interfaz gráfica sencilla y clara, mostrando las funcionalidades principales utilizadas por los usuarios.

Una de las principales características de **Gedit** es que además de tener las funciones del típico procesador de textos, se puede desarrollar archivos de programación en distintos lenguajes y trabajar con varios archivos a la vez mediante pestañas.

3 Arquitectura del sistema

Después de realizar un análisis exhaustivo de los requisitos de nuestro sistema, se concretó la arquitectura del sistema de la siguiente manera. A continuación, se explica cada uno de los componentes de la arquitectura.

3.1 Servidor de Telegram

En el servidor de Telegram se recogen todos los mensajes que los usuarios del bot le envían a éste. Una vez han llegado al servidor de Telegram, se procede a la lectura de los mensajes. Telegram proporciona al propietario de un bot un TOKEN asociado a ese bot. De esta manera, podemos rescatar todos los mensajes de nuestro bot, de manera sencilla y segura.

Cada mensaje tiene una lista de atributos, entre ellos podemos encontrar, por ejemplo, el nombre de usuario, el identificador del chat, la fecha del mensaje o el texto contenido en el mensaje.

3.2 Servidor de la universidad

En el servidor de la universidad es donde se tratan los mensajes. Mediante el uso del TOKEN, se recuperan los mensajes del servidor de Telegram. La identificación del contenido del mensaje se facilita mediante el uso de comandos. Cada acción del bot está asociada a un único comando, de esta manera podemos identificar de manera sencilla la acción que quiere realizar el usuario.

La aplicación corre en un único proceso, y es éste proceso el que realiza diferentes tareas. Por un lado, implementa un delegator. Este delegator se encarga de proporcionar tres hilos a cada usuario.

Un hilo se reserva para el envío constante de noticias, de esta manera, no bloqueamos la llegada de nuevos comandos. Un segundo hilo trata los nuevos mensajes, así, cada vez que un usuario se comunica con el bot, ejecuta sus acciones en el hilo proporcionado para ello. El tercer hilo se encarga de gestionar los “inline query”. Con el comando /discoverFeeds se proporcionan unos botones al usuario que cuando son pulsados, envían un mensaje distinto al que aparece en el botón, de esta manera, podemos hacer que el uso del bot sea más simple para usuarios con dificultades en el manejo de las nuevas tecnologías.

Para tener un control de la información de cada usuario, se hace uso de una base de datos relacional instalada en el propio servidor. Esta base de datos está compuesta por tres tablas. La primera tabla, Feed, almacena los feeds que ha agregado cada usuario. La segunda tabla, Noticia, almacena la url de la noticia y el usuario al que ha sido enviada, de esta manera, evitamos que se repita el envío de noticias por parte del bot, a un usuario. La tercera tabla, Sugerencia, almacena la similitud de Jaccard entre los distintos feeds.

Además, el proceso mencionado anteriormente, se encarga de realizar otras tareas. Por un lado, se encarga de actualizar la programación de la televisión. Mediante web scraping obtenemos la programación de cada canal de televisión y lo guardamos en un diccionario que será consultado cada vez que ejecutemos el comando asociado a la programación. Además, por último, se encarga de evaluar las similitudes entre los distintos feeds que están contenidos en la base de datos.

3.3 Clientes

Es el que inicia la conversación con el bot. Si un cliente no envía previamente un mensaje al bot, el bot no podrá escribir al usuario. Después de haber enviado un primer mensaje, el bot tiene vía libre para escribir al cliente.

El cliente dispone de una serie de comandos que facilitan el uso del bot. Por ejemplo, el cliente enviará el comando principal de agregar un feed, y una vez que el bot haya recibido el mensaje, lo añadirá a la base de datos, y será el bot el que conteste con sucesivas noticias de ese feed. En este caso, se trata de un único tipo de cliente por lo que el trato con los clientes es siempre igual.

4 Descripción del sistema

En este apartado, vamos a describir el funcionamiento del sistema, sus aplicaciones y requisitos como cliente y servidor. Analizaremos detalladamente la aplicación.

4.1 *Ámbito del sistema*

Nuestro objetivo principal, es desarrollar un bot en la plataforma de mensajería Telegram.

El bot es una aplicación online, usamos la aplicación de Telegram para mandar la información a los usuarios.

El propio programa es el encargado de gestionar los hilos para cada usuario. Existen hilos paralelos para actualizar la información, que, posteriormente, será enviada a los usuarios.

4.2 *Funcionalidades de la aplicación*

Al arrancar el bot en el servidor, éste carga los datos de la programación de televisión de ese día. Los datos se guardan en una variable del programa. El tipo de variable es un diccionario, donde la clave es el canal de televisión y el valor una cadena con la programación del día. Todo ello se ejecutará en un hilo independiente, el cual, mediante un Timer, actualizará la información cada 6 horas.

Una vez cargada esta información, el programa se quedará a la espera de que algún usuario interactúe con él.

Cuando un bot es invocado (buscándolo en el buscador de contactos o invocándolo en las conversaciones, mediante el comando **@Read_News_Bot**), el programa generará tres hilos para cada usuario. En un primer hilo, se guarda información del usuario que está interactuando, como puede ser el nombre de usuario, su identificador de Telegram, etc. Esta información, es guardada en una base de datos SQL.

A continuación, se muestra un mensaje de bienvenida, y la información de los comandos con los que el usuario puede comunicarse.

Si el usuario selecciona el comando */agregarfeed enlaceDelFeed*, se añadirá el feed a la BBDD. Posteriormente, realizaremos comparaciones de ese feed con el resto de

feeds, de manera que si encontramos un feed parecido, sugerimos al usuario que agregue ese feed.

Cuando el usuario tenga algún feed agregado, el otro hilo leerá las noticias proporcionadas por el feed que tiene agregado y si no las ha enviado, procederá a enviarlas.

4.3 Exploración de la aplicación

En esta sección, se van a ver las distintas funcionalidades del bot. Se explicarán detalladamente los pasos a seguir para su uso en los distintos ámbitos del bot.

En primer lugar, se busca el bot en el buscador principal de Telegram. El bot, interactúa de la misma manera que una persona. Se busca por su nombre **@Read_News_Bot**.

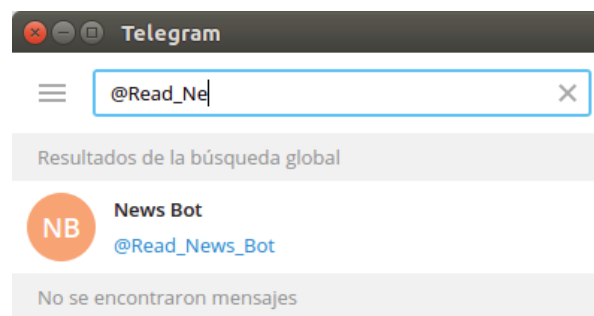


Figura 12. Buscando el bot

Cuando se haya encontrado, se inicia pulsando sobre él. A continuación, aparecerá una ventana en blanco y con un único botón *start*, donde aparece una descripción del bot.

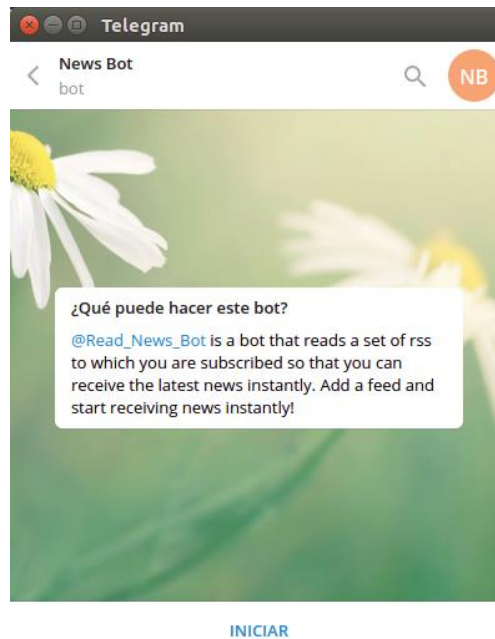


Figura 13. Arrancar el bot

Al seleccionar `/start`, el bot mandará un mensaje personalizado de bienvenida y mostrará información sobre su funcionalidad.



Figura 14. Bienvenida del bot

Una vez iniciado, en el teclado aparece el caracter `/`, el cual, si es pulsado, se muestran los comandos de la aplicación.

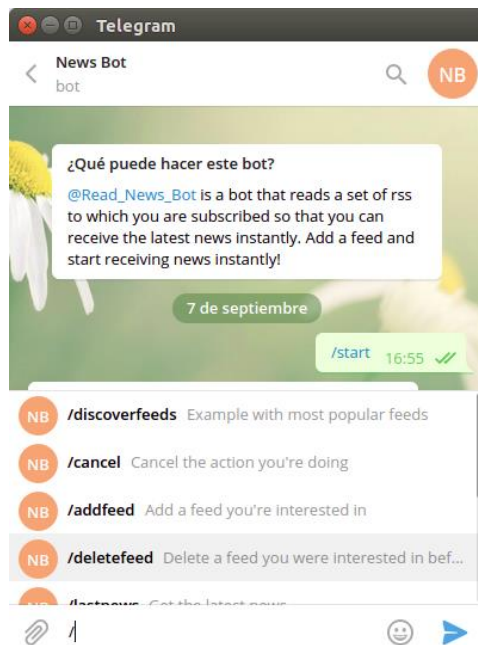


Figura 15. Comandos 1

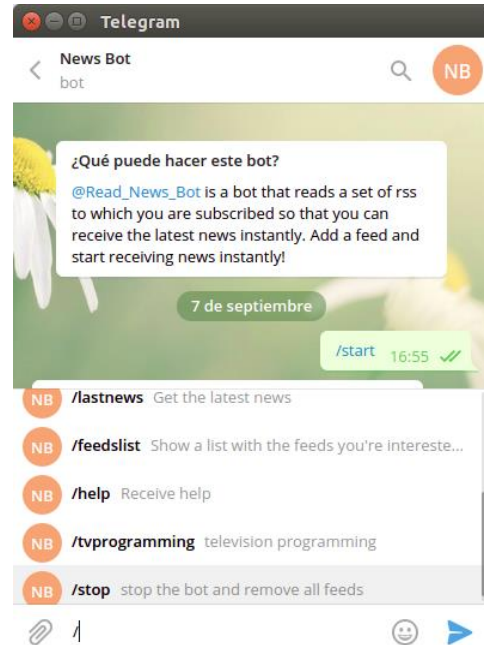


Figura 16. Comandos 2

A continuación, se analiza la funcionalidad de cada comando.

/help

Muestra un mensaje con los comandos del bot y su funcionalidad.

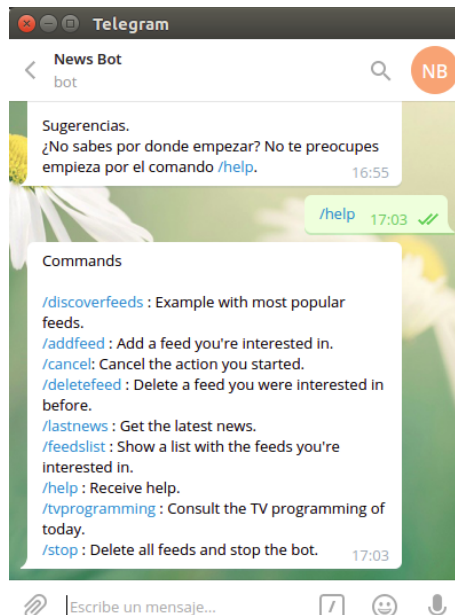


Figura 17. Comando ayuda

/tvprogramming

Al seleccionar este comando, el bot muestra una lista de botones (ordenados alfabéticamente) de todos los canales de televisión que tiene el TDT.

Una vez elegido el canal, el bot envía la programación de televisión del día actual.

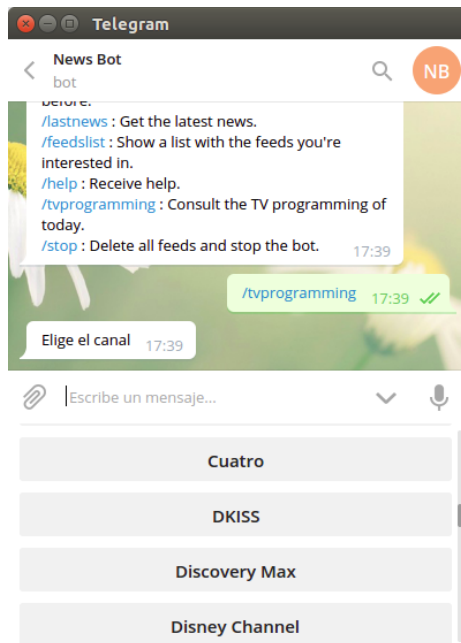


Figura 18. Canales de tv

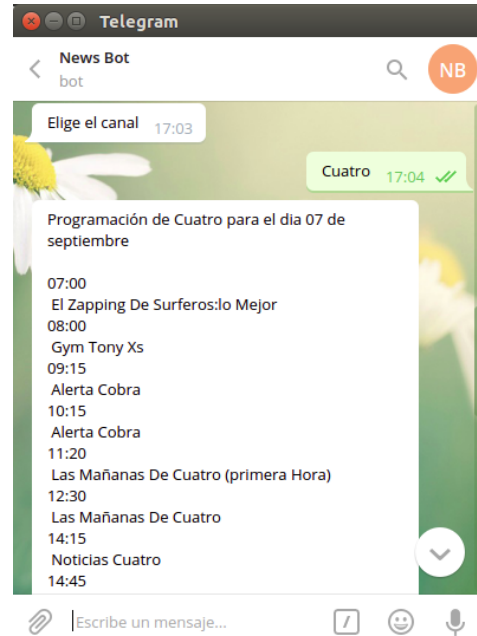


Figura 19. Programación tv

/addfeed “RSS”

Junto con el comando /discoverfeeds, es el comando encargado de realizar la función más importante del bot.

Se trata de un comando que incluye el RSS de las noticias que le interesan al usuario. Se pueden buscar los feed.



Figura 20. Agregando un feed



Figura 21. Respuesta /addfeed

/discoverfeeds

Este comando ofrece la posibilidad de agregar feed. Proporciona al usuario, los feed de dos periódicos nacionales y otros dos deportivos(en forma de botones).

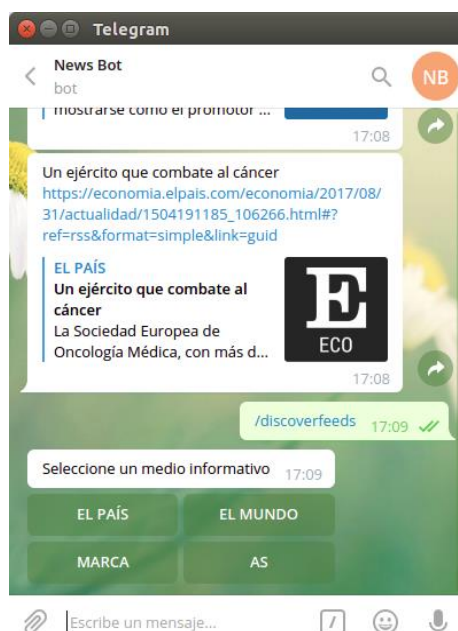


Figura 22. Periódicos

Una vez seleccionado alguno de lo periódicos, el usuario recibe una lista de botones según la temática, al seleccionar alguno de ellos, se añade el feed automáticamente al usuario.

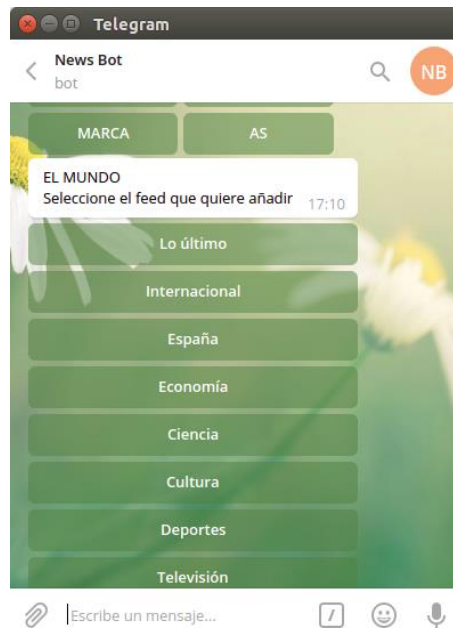


Figura 23. Temas a agregar

/feedslist

Muestra una lista con los feeds que tienes agregados



Figura 24. Lista de feeds

/deletefeed

El bot ofrece los feeds añadidos por el usuario en forma de botones. Cuando es seleccionado alguno de ellos, se borra del perfil del usuario y deja de recibir noticias de ese feed.

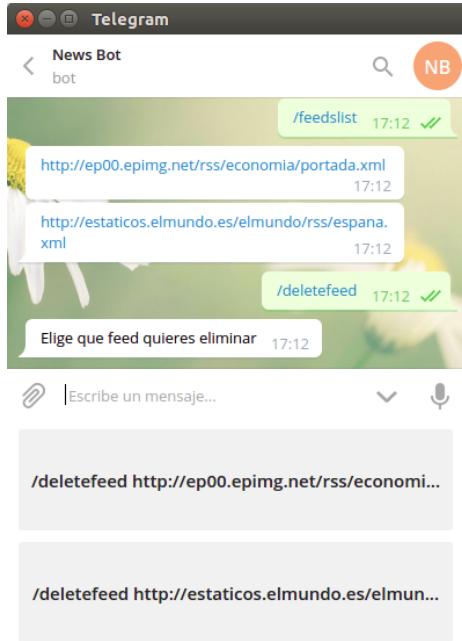


Figura 25. Botones eliminar feed

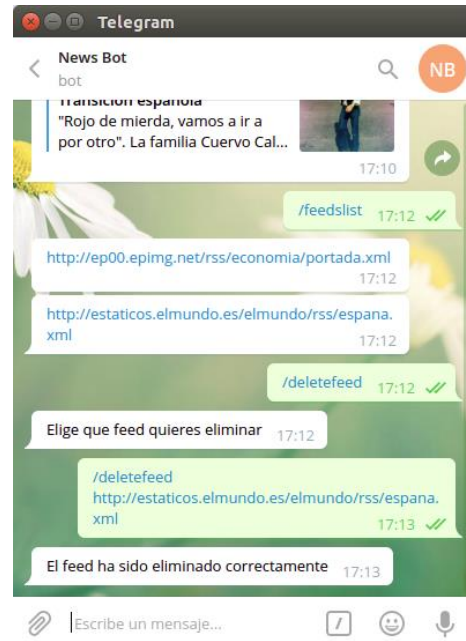


Figura 26. Borrado confirmado

/lastnews

Al seleccionar este comando, el bot muestra una lista de fuentes. Después de escoger la fuente, el bot envía las últimas noticias de esa fuente.



Figura 27. Últimas noticias

/cancel

Se interrumpe la acción que el usuario ha seleccionado.



Figura 28. Cancelar acción

/stop

Detiene los hilos propios del usuario y además da de baja de los feeds que tenía añadidos.



Figura 29. Parar el bot

La mayor característica del bot, es el **envío de noticias**.

Cuando se añade un feed, el bot envía todas las noticias que hay en el RSS que se ha elegido. Una vez enviadas, cada una hora el programa actualiza la BBDD, añadiendo las noticias nuevas desde el último envío. Comprueba las que no le ha enviado al usuario y se las envía.



Figura 30. Envío de noticias.

Por otro lado, la aplicación también incluye un **recomendador** de RSS.

Se pensó en realizar un recomendador para mejorar la experiencia final del usuario, basándonos en el comportamiento del resto de usuarios. Para realizar el recomendador, de acuerdo con algunas técnicas vistas en la carrera, utilizamos la similitud de Jaccard como métrica de distancia entre los feeds.

La similitud de Jaccard¹ de dos conjuntos S y T es $|S \cap T| / |S \cup T|$, es decir, la relación entre la intersección de los conjuntos S y T , y la unión de éstos.

A la hora de calcular la similitud de Jaccard entre los distintos feeds que tenían agregados nuestros usuarios, formamos una matriz que toma valores binarios. Tomará el valor '1' en caso de que un usuario 'U' tenga agregado el feed 'F', y en caso contrario tomará el valor '0'. Tomaremos como ejemplo la siguiente tabla.

¹ De acuerdo con el apartado 3.1.1 Jaccard Similarity of Sets, donde se define la similitud de Jaccard

<http://infolab.stanford.edu/~ullman/mmds/book.pdf>

F\U	U1	U2	U3	U4
F1	0	0	1	1
F2	1	0	1	1
F3	0	1	0	0
F4	0	1	1	1

Aplicando la similitud de Jaccard a los conjuntos formados por los valores que recoge la tabla de los feed 'F1' y 'F2', se obtiene la conclusión de que la similitud entre ambos conjuntos es:

$$|F1 \cap F2| / |F1 \cup F2| = 2 / 3$$

De esta misma manera, evaluamos el resto de relaciones entre dos pares distintos de feeds.

$$|F1 \cap F3| / |F1 \cup F3| = 0 / 3$$

$$|F1 \cap F4| / |F1 \cup F4| = 2 / 3$$

$$|F2 \cap F3| / |F2 \cup F3| = 0 / 4$$

$$|F2 \cap F4| / |F2 \cup F4| = 2 / 4$$

$$|F3 \cap F4| / |F3 \cup F4| = 1 / 3$$

Observando el feed 'F3', llegamos a la conclusión de que no se asemeja a los feed 'F1' y 'F2', puesto que el usuario que tiene agregado el feed 'F3' no muestra interés por los feeds 'F1' y 'F2', y viceversa. En cambio, se puede observar que los feeds 'F1' y 'F2' sí se asemejan, puesto que varios usuarios que tienen un feed, tienen además el otro. De esta manera, si un usuario agrega uno de los feeds, en este caso 'F1' o 'F2', se le sugerirá que agregue el otro feed.



Figura 31. Recomendador

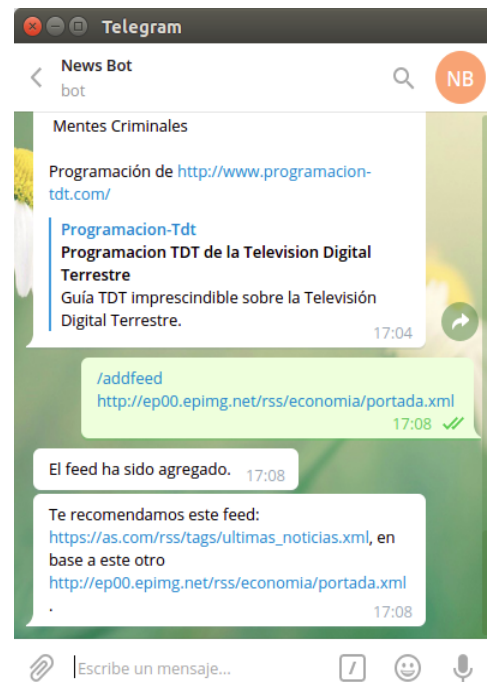


Figura 32. Recomendando feed

4.4 Requisitos

En este apartado, se explican los elementos necesarios para poder ejecutar la aplicación. Por un lado están, los requisitos del cliente:

- Disponer de un usuario de Telegram.
- Aplicación de Telegram actualizada a la última versión.
- Conexión a internet

Por otro lado, los requisitos del servidor, los cuales necesitan la instalación de los siguientes elementos:

- Python 2.7 o posteriores
- Librerías de Python (telepot, BeautifulSoup...)
- MySQL Service

Por último, un elemento fundamental es que los RSS que se añaden, estén disponibles, al igual que la web de donde se saca la información de la programación de televisión (<http://www.programacion-tdt.com/>).

5 Puntos críticos del proyecto

5.1 Ley de protección de datos

Al principio del proyecto nos planteamos si nuestro bot se podía llevar a cabo y el principal motivo fue porque estábamos desarrollando un bot que es un lector RSS que lee noticias que están publicadas en esos RSS. Cada periódico tiene su propio RSS y realizamos un estudio sobre los términos y condiciones de cada periódico y los derechos de autor.

Nos informamos y leyendo leyes encontramos un punto clave en nuestro proyecto:

“4.4.1. Uso personal de los titulares. El usuario podrá utilizar los titulares en su computadora personal (mediante el uso de programas de terceros que emplean ficheros RSS para mostrarle tickers de noticias) sin ningún tipo de retransmisión de los mismos.”

En conclusión, gracias a este punto pudimos empezar a desarrollar nuestro bot, siendo un lector de RSS dentro de la plataforma Telegram.

5.2 Noticias repetidas

Cuando agregábamos un feed, y el bot nos enviaba noticias, nos dimos cuenta que en ocasiones, nos enviaba una noticia que ya la habíamos recibido. Nos dimos cuenta, que los periódicos a veces modifican el título de las noticias. La noticia se enviaba duplicada, porque en la BBDD guardamos el título como clave primaria de la tabla.

Lo corregimos cambiando la clave primaria. Pusimos el enlace a la noticia como clave y no el título, ya que el enlace se seguía manteniendo aunque se modificara el título.

5.3 Problemas con tiempos de cómputo

La creación de un bot, viene relacionado con el tiempo de respuesta, es una de los puntos principales en el proyecto. Hemos intentado que el usuario no apreciara el tiempo de respuesta, desde que envía un comando hasta que el bot le contesta con lo que el usuario le ha pedido. Intentamos buscando los componentes que eran los más adecuados, como MySQL, el manejo de esta base de datos de tipo relacional ha sido clave para guardar los datos.

Además, en la hora del desarrollo tuvimos problemas a la hora de arrancar el servidor, tardaba mucho tiempo en correr el bot, por lo que tuvimos que hacer mejoras de código en las consultas a la base de datos y la estructura del comando programación.

5.4 Lenguaje Python

El principal problema que tuvimos al empezar proyecto, fue decidir en qué lenguaje lo íbamos a desarrollar.

Nunca habíamos implementado algo de este estilo, por lo que estábamos un poco perdidos.

En una de las reuniones con nuestro tutor Carlos, nos propuso que lo desarrolláramos en Python. Nos habló de una librería que él conocía (telepot) y nos podría ayudar. Ninguno de los miembros del equipo, había trabajado con Python, por lo que nos planteamos si estábamos dispuestos a empezar de 0 un lenguaje para desarrollar el TFG y el trabajo que ello supondría.

Realizando algunos bots sencillos, nos dimos cuenta que no nos resultaría tan difícil aprenderlo, por lo que decidimos implementarlo con él.

5.5 Botones de Telegram

A la hora de mostrar todos los canales de televisión, al ser un número elevado de ellos (63), no conseguimos mostrar todos, se nos cortaban los botones.

El `ReplyKeyboardMarkup()` hay que pasarle como parámetro, un array de un array. Funciona como una matriz. Nuestro problema era que solo mostraban los botones que cabían en el cuadrado que ve el usuario. Encontramos el error, viendo que el parámetro que le pasábamos, estaba mal (le pasamos un único array con la lista de botones), la solución fue meter ese array dentro de otro array.

5.6 Hilo actualizar programación

La programación de televisión, tenemos que actualizarla cada día, ya que cambia de un día para otro. Pero luego nos dimos cuenta de que podía ocurrir, que se cancelara algún evento durante el día y la programación cambiase. Decidimos actualizarlo cada 6 horas.

5.7 Entrega en junio

Empezamos el proyecto con muchas ganas en octubre, informándonos y buscando información sobre los bots de Telegram. Con el paso de los meses, lo fuimos aparcando debido al gran número de asignaturas que teníamos en el primer cuatrimestre. El segundo cuatrimestre pudimos dedicarle más tiempo porque teníamos menos asignaturas. El problema fue que empezamos a trabajar y se volvió a complicar, entre el poco tiempo y el aprender un nuevo lenguaje de programación (ninguno de los miembros, habíamos programado en Python). Sobre marzo, decidimos dejar casi terminada la parte funcional del bot y dedicar el verano a mejorar la funcionalidad y terminar la memoria.

6 Conclusiones del proyecto

Durante el desarrollo del proyecto, hemos podido participar en una rama de la informática que ha cobrado mucha importancia en los últimos años y sobre todo, en cuanto a las aplicaciones de mensajería se refiere.

Actualmente, los bots están entrando con fuerza en el mundo de la tecnología, por ello, pensamos en esta aplicación para mostrar los contenidos aprendidos a lo largo de la carrera. Por otro lado, como ocurre en el mundo laboral, hay que adaptarse a las herramientas y software con las que se trabajan, y no siempre se tienen conocimientos sobre ellas. Durante el proyecto, nos pusimos el reto de desarrollarlo en un lenguaje como Python, el cual, nunca habíamos utilizado. Podemos decir a día de hoy que nuestros conocimientos de Python han aumentado considerablemente, siendo un lenguaje muy eficaz para la extracción de datos.

La primera propuesta y la única que nos motivó a elegir este proyecto era desarrollar un bot relacionado con los deportes y con páginas de apuestas deportivas, el cual pudiese ser utilizado por todos los usuarios. Debido a su complejidad, apartamos esa idea inicial y nos hicimos una pregunta, ¿por qué no realizar un bot que se pareciera a un periódico tradicional?. La idea nos pareció perfecta ya que con un simple bot, podíamos estar informados de los temas de actualidad que eligiéramos.

A la hora la extracción de noticias, nos dimos cuenta de la cantidad de webs que ofrecen un servicio RSS y lo útil que puede llegar a ser. Este servicio, evita conflictos con la ley de protección de datos.

Por otro lado, hemos utilizado los conocimientos aprendidos durante la carrera en base de datos a la hora de guardar la información con MySQL.

Tenemos que destacar, el uso de la librería para Python telepot, una API para desarrollar Bots en Telegram, facilitándonos la creación de la aplicación.

Por otro lado, gracias a la universidad, hemos podido hacer numerosas pruebas, ya que nos ha proporcionado un servidor. También lo hemos usado como repositorio Git para poder trabajar coordinados.

Aunque durante estos años de carrera hemos adquirido experiencia y conocimiento, el desarrollo del proyecto ha sido una gran prueba a nivel individual y colectivo. El gran número de pruebas realizadas, los errores y puntos críticos que han aparecido a lo largo del proyecto, nos han servido para abarcar situaciones que viviremos día a día en el mundo laboral.

Tras los meses de trabajo, los objetivos se han cumplido y esperamos que el bot atraiga a numerosos usuarios de Telegram.

6.1 Project's conclusions

During the development of the project, we have been able to participate in a branch of computing that has become very important in recent years and especially, as far as messaging applications are concerned.

Currently, the bots are entering with force in the world of the technology, for that reason we think in this application to show the contents learned throughout the career. On the other hand, as in the working world, you have to adapt to the tools and software you work with, and you don't always know about them. During the project, we set ourselves the challenge of developing it in a language like Python, which we had never used. We can say today that our python knowledge has increased considerably, being a very effective language for the data extraction.

The first and only proposal that motivated us to choose this project was to develop a bot related to sports and sports betting pages, which could be used by all users. Due to its complexity, we put aside that initial idea and we asked ourselves a question, Why we don't make a bot that looks like a traditional newspaper?. The idea seemed perfect because with a simple bot, we could be informed of current topics that we chose.

At the time of extracting news, we realized the number of websites that offer an RSS service and how useful it can become. This service avoids conflicts with the data protection law.

On the other hand, we used the knowledge learned during the career in database when saving the information with MySQL.

We have to highlight the use of the library for python, Telepot, an API to develop Bots in Telegram, making easy the creation of the application.

On the other hand, thanks to the university, we have been able to do some tests, since we have a server. We have also used it as a Git repository to work in coordination.

Although during these years of career we have acquired experience and knowledge, the development of the project has been a great test at individual and collective level. The large number of tests performed, the errors and critical points that have appeared throughout the project, have served us to cover situations that we will live day by day in the working world.

After months of work, the objectives have been fulfilled and we hope that the bot will attract numerous users of Telegram.

7 Ampliaciones futuras

Tras el desarrollo del proyecto se proponen algunas funcionalidades que se podrían añadir al bot en un futuro:

1. Desarrollar parte del bot para la lectura de RSS para redes sociales (Twitter, Facebook, Instagram, YouTube).
2. Otro punto que se puede añadir al bot desarrollado es que los usuarios puedan interactuar entre ellos, comentando los feeds o noticias.
3. El comando agregarFeeds se podría mejorar con un buscador de feeds. Solo con introducir un tema que busque diferentes RSS. Un ejemplo es introducir el tema fútbol y que muestre una lista de feeds que sean de fútbol, exponiendo las más utilizadas en todo el mundo.
4. En un futuro queríamos emplear el componente big data ya que recoge gran cantidad de información al igual que el proyecto desarrollado.

7.1 Recomendador

Un recomendador puede tener multitud de información acerca de un usuario, de una noticia, o de un feed. Se podría desarrollar un recomendador que recomiende noticias, basado en las evaluaciones de otros usuarios. A medida que se recolecta información, más preciso podría llegar a ser el recomendador. El recomendador que se ha desarrollado podría avanzar en ese sentido. Además, el recomendador podría hacer comprobaciones sobre el propio usuario, como por ejemplo, sugerir un feed si y solo si no tiene ese feed agregado.

7.2 Puntuación / comentarios de noticias

Una de las ampliaciones, sería la de hacer más sociable el bot. Si los usuarios puntúan las noticias, se puede valorar la noticia según su puntuación.

Por otro lado, no estaría de más dar la funcionalidad de comentar una noticia. Se llevaría a cabo a través de un botón enlazado a la noticia.

La mejora se resume; al recibir una noticia, ésta vendría con 3 botones, uno puntuar, otro para comentar y por último otro para ver los comentarios de esa noticia.

7.3 Programación de TV

Al mostrar la programación de un canal, el usuario podrá indicar qué es lo que va a ver, y se guardará en su BBDD. Posteriormente, el usuario podrá ver una lista con todo lo que ha visto esa semana.

Estos datos, se usarán para que un recomendador, te diga qué ver cada día.

8 Contribuciones personales

En este apartado cada participante va a hablar sobre sus aportaciones al proyecto, detallando su participación y las ayudas ofrecidas al resto de compañeros.

Se detallará todo en las siguientes partes: Conocimientos, Investigación, Pruebas, Desarrollo de la Aplicación y Memoria.

8.1 Andrés Aguirre Juárez

8.1.1 Conocimientos

Como estudiante del grado de ingeniería informática, he adquirido conocimientos de programación en distintos lenguajes, programación de algoritmos. Además tengo conocimientos de programación web y manejo de bases de datos.

8.1.2 Investigación

Después de un año como usuario de Telegram, encaré la recta final de la carrera buscando los trabajos de fin de grado propuestos por los profesores. Este proyecto, fue el que más me llamó la atención, puesto que significaba desarrollar un bot sobre una plataforma con millones de usuarios activos.

Estuve investigando en verano, antes de la primera reunión con nuestro tutor, y había pensado en desarrollar el bot en PHP, puesto que era una de las sugerencias de Telegram y además yo tenía conocimientos de PHP. Después de la primera reunión con nuestro tutor, nos sugirió que utilizáramos Python. Estuve investigando porque no conocía Python y después de una semana de reflexión, vi más aspectos positivos que negativos a la hora de utilizarlo, puesto que iba a aportarme conocimientos en un lenguaje potente y muy utilizado en la actualidad.

8.1.3 Pruebas previas

Las primeras pruebas que realicé, estaban desarrolladas con la API que nos sugirió nuestro tutor en la primera reunión del proyecto. Esta API, me permitió desde un primer momento acercarme al desarrollo del bot de una manera muy sencilla. La API es tan completa que al final acabamos desarrollando con ella.

El primer acercamiento que tuve con los bots, fue desarrollando un bot que en base a una función random, después de preguntarle algo, te contestaba sí o no. Este primer bot, en realidad, contestaba a cada mensaje con sí o no en base a esa función random.

Más adelante, realicé otro bot que te devolvía todos los enlaces que encontrase en una página web que le enviasen. De ahí surgió la idea de realizar un bot relacionado con noticias.

Por último, antes de comenzar con el desarrollo de nuestro bot, las pruebas previas que realicé, consistían en ejecutar códigos de ejemplo de algunos bot creados, introduciendo pequeñas modificaciones, para comprender el correcto funcionamiento de esos bots que habían sido desarrollados por otros usuarios.

8.1.4 Desarrollo de la aplicación

En primer lugar, creé el bot. Para crear un bot en Telegram, lo que hay que hacer es hablar con otro bot, BotFather. Después, comencé con la primera versión de la lógica que debía utilizar nuestro bot. Implementé varios comandos como `addfeed` o `feedslist`. Estuve investigando y descubrí que los periódicos disponen de un servicio RSS mediante el que publican sus noticias.

A través de la librería `feedparser` para Python, estuve desarrollando la funcionalidad de enviar una noticia. Primero realizo una lectura de los títulos de las noticias y la URL y después paro el hilo un tiempo, hasta que vuelven a leerse las noticias.

Por último, me encargué del desarrollo del recomendador de feeds. Basándome en lo aprendido en la asignatura de minería de datos que cursé en la carrera, realicé un recomendador que en base a la experiencia de otros usuarios, te recomienda un feed nada más añadir otro.

8.1.5 Memoria

A la hora de redactar la memoria, una de mis funciones ha sido revisar las aportaciones de cada miembro del grupo. Esas revisiones consistían desde evaluar si el lenguaje utilizado era el apropiado, hasta evitar que realizáramos aportaciones sobre los mismos aspectos del proyecto en distintos capítulos. Además, he realizado el capítulo de la arquitectura del sistema.

Por último, en el capítulo 5 sobre la descripción del sistema, he redactado el funcionamiento del algoritmo de recomendación, así como un breve ejemplo para comprender mejor el funcionamiento del algoritmo.

8.2 Diego Reguilón Rodríguez

8.2.1 Conocimientos

Como estudiante del grado de ingeniería de computadores, he adquirido conocimientos de programación en distintos lenguajes y manejo de base de datos. En mi carrera profesional, me dedico al sector del Big Data.

8.2.2 Investigación

Hablando con mis compañeros, uno de ellos me comentó que se ofertaba este proyecto como TFG. Al ver que Telegram era la plataforma de mensajería elegido, mi interés en el proyecto aumentó.

Soy usuario de Telegram, casi desde sus orígenes. De hecho, poco después de entrar en la universidad, empezamos a utilizarlo con compañeros de la carrera. Conocía su amplio abanico de funcionalidades, entre ellas, me llamó especial la atención los nuevos bots.

Ante mi expectación sobre los bots, decidí aceptar este proyecto como TFG.

Empecé a investigar sobre los bots de Telegram y las mejores APIS para su desarrollo. Siguiendo la recomendación de nuestro tutor, nos comentó que existía una para Python llamada **telepot**, empezamos a investigar sobre ella.

Es una API desarrollada por un usuario de Python llamado @nickoala.

Nuestro tutor, nos comentó que si usabamos linux como sistema operativo, nos podría ayudar más, ya que nos iba a ser más útil.

Hice una bipartición de disco y añadí Ubuntu a mi ordenador. Me instalé Python (ya que no lo había utilizado nunca) y realicé algunos ejemplos sencillos donde poco a poco fui aumentando la dificultad.

Continué investigando, estudiando cómo funcionaba telepot y sus diferentes servicios, realizando varios ejemplos con él.

Por otro lado, tuve la necesidad de parsear una página web, y buscando, encontré una librería llamada **Beautiful Soup** para Python, cuya finalidad es analizar documentos HTML para extraer su información.

También me tocó lidiar con los Threads en Python, al haberlos usado solo en C, me tuve que adaptar a ellos. Realicé algunos ejemplos sencillos que me fueron útiles para el proyecto.

Al principio del curso, donde aún no trabajaba, empecé a investigar un poco acerca del servidor que nos propuso nuestro profesor, nunca había trabajado con ellos, ni con las autenticaciones ssh. Me di cuenta que no tenía mucha dificultad.

A mitad de curso, cuando ya trabajaba y usaba clusters con varios nodos, dominaba a la perfección esta tarea.

8.2.3 Pruebas previas

Las primeras pruebas que hicimos, fueron muy simples, mantener una comunicación con el bot, de forma que viera lo que le mandabas y el bot fuera capaz de contestarte.

Continuamos las pruebas con un bot simple, donde se realizaba un sistema de votación para elegir los mejores jugadores de fútbol, mediante botones, se contabilizaba el número de votos y se podían ver los resultados. El bot te mandaba una foto al instante de votar, del jugador que habías elegido y su porcentaje total de votos.

Luego realizamos un pequeño bot que guardaba datos de usuarios, y los votos que realizaban en una base de datos SQL. El manejo de BBDD SQL con Python es muy sencillo.

La última prueba que realicé, fue parsear un XML sencillo, y extraer su información (el xml del proyecto, era bastante más complicado). Paseábamos noticias de webs, pero nos dimos cuenta que los RSS de los periódicos, nos serían mucho más útiles que parsear distintas webs.

8.2.4 Desarrollo de la aplicación

Después de hablar con nuestro tutor y acordar la temática del bot, hicimos una reunión para montar la arquitectura de la aplicación. En ella dividimos el trabajo que haríamos.

La aplicación tiene dos partes de la lógica bien diferenciadas, por un lado las lecturas y tratado de los RSS y por otro lado la programación de televisión. Yo me he encargado de hacer esta última parte.

La funcionalidad es clara, ofrecer al usuario una lista de canales (63), que ofrece el TDT (todo gratuitos). Al seleccionar el canal, el bot te envía la programación para el día actual.

Empecé a investigar librerías para extraer la información de una página web. Buscando, di con *Beautiful Soup*, la cual crea un árbol con todos los elementos del documento y puede ser utilizado para extraer información. Mediante filtrados, conseguí extraer los canales de TV.

Con otro parseo diferente, extraí la programación que ofrecía cada canal. Esta información se guarda en un diccionario, donde la clave es el canal y el valor su programación.

Cada día cambia la programación, por lo que mínimo había que lanzar el proceso una vez al día para actualizar los canales.

Busqué información sobre los hilos en Python, y encontré una función del *threading* de Python, donde llamas a la función *Time* y le pasas como argumentos, un entero (segundos del timer), y una función. Funciona de tal manera, que el hilo llama a la función que le pasas, cada X tiempo. Siento X el número de segundos que espera el hilo hasta volver a ejecutar la función.

En cuanto a la parte de noticias, desarrollé el comando */discoverfeeds*. Al seleccionar el comando, el bot envía cuatro botones de diferentes periódicos. Estos botones son enviados como un mensaje dentro de la pantalla en forma de mensaje (*InlineKeyboardMarkup*). Una vez seleccionado el periódico, el bot te envía una lista de botones (del mismo tipo que los anteriores), con distintas secciones del periódico. Cada botón tiene un valor asociado con el enlace del feed correspondiente (valor del *callback_query*). Al seleccionar un botón, el programa invoca al comando */addfeed*, pasándole como argumento el valor del botón (feed a agregar) y el feed es agregado al usuario.

Por otro lado, me he encargado de los controles de errores de la aplicación. Junto con mis compañeros, realizamos el menú de los comandos, así como la gestión de los hilos por cada usuario.

Por último, he realizado la subida al servidor de la aplicación, ya que me encargaba de esas gestiones dentro del equipo.

8.2.5 Memoria

En esta parte del proyecto, me he encargado de realizar el resumen del proyecto, enlazándolo con el auge tecnológico que estamos viviendo.

También he realizado la introducción del proyecto, y la Descripción del sistema. He estado muy pendiente de lo que desarrollaban mis compañeros (también les ayudaba) y es por eso que conozco muy bien el funcionamiento del sistema. Se me da bien explicarme y para esta parte es fundamental, ya que el cliente tiene que comprender y entender bien la aplicación.

Por otro lado, también he contribuido en los problemas y soluciones que hemos encontrado durante el proyecto, de igual forma que en las ampliaciones futuras. En estos dos puntos también han participado mis compañeros.

8.3 Adrián Román Urdiales

8.3.1 Conocimientos

Soy estudiante del Grado en Ingeniería de Computadores. He adquirido durante la carrera conocimientos en lenguajes de programación como java, C, C++ y selección de componentes tecnológicos.

Actualmente, estoy trabajando en el departamento de servicios Transaccionales y Tecnología como arquitecto de aplicaciones y soluciones.

8.3.2 Investigación

Lo primero que me llevó a elegir el Trabajo de Fin de Grado fue el poder de desarrollar un bot en la aplicación de mensajería Telegram. Soy usuario desde el lanzamiento de la aplicación y quería aplicar mis conocimientos adquiridos.

La primera reunión con nuestro tutor me hizo ver que es una aplicación que no tiene límites y que podía desarrollar cualquier tipo de bot que tuviera en mente. Empecé a buscar información sobre los bots que nos proporciona Telegram.

Tras llevar a cabo un estudio detallado se observó que se podía desarrollar el bot en cualquier tipo de lenguaje. Como la mayoría de los bots estaban desarrollados en Python y PHP, nos centramos en lenguaje de programación Python. Éste fue el lenguaje que nos recomendó nuestro tutor. Además, nos sugirió el API Telepot: desarrollada por Nick Lee (@nickoala), se trata de una API bastante interesante para el desarrollo de bots.

Tras varias reuniones con mis compañeros de grupo, se estudió que bot era el adecuado para plasmar los conocimientos de la carrera y nos decidimos por realizar un bot para que los usuarios pudiesen tener información en tiempo directo.

Además, se estudiaron los diferentes componentes tecnológicos que han sido importantes para el desarrollo de la aplicación como MySQL, Python y RSS.

8.3.3 Pruebas previas

Las primeras pruebas con bots de Telegram fueron con los lenguajes de programación en PHP y Python. En primer lugar pensamos en PHP para el desarrollo de nuestro bot. Se realizaron una serie de pruebas con distintos bots y se observó que no era adecuado para nuestro proyecto. Por tanto nos decantamos por utilizar *Python*, desarrollando bots sencillos como por ejemplo, un bot que generara un número aleatorio

de 0 a 100 y tenía que acertar el número elegido. Además, probé los ejemplos propuestos por mis compañeros y los que aparecen en el repositorio de @nickoala.

Una vez que realizamos las pruebas en bots, llevamos a cabo pruebas en BBDD, ya que para nuestro proyecto la característica que tenía que tener era la velocidad y que no hubiera pérdida de datos. Por esta razón utilizamos MySQL.

8.3.4 Desarrollo de la aplicación

Antes de empezar a desarrollar, nos reunimos para ver los diferentes componentes elegidos y repartirnos las tareas. Se definieron las tablas que contiene nuestra BBDD.

Con mi compañero Andrés me he encargado del desarrollo del lector de RSS y en el comportamiento de leer los RSS de los usuarios.

Al dividirnos trabajo tuve que encargarme de la creación de hilos para los diferentes usuarios que utilizan nuestra aplicación. Al principio pensamos en la creación de Threads estudiados en la carrera, pero al utilizarlos me di cuenta que no tenía control de los Threads en los usuarios. Buscando información, mi compañero Andrés encontró dentro de la API Telepot el componente *DelegatorBot*.

Estudí cómo se comportaba y lo apliqué al desarrollo de nuestro bot.

Además, me responsabilicé en buscar soluciones a los problemas que nos iban surgiendo, como por ejemplo el tiempo de cómputo de nuestro proyecto.

También desarrollé un ejecutable que recogía los mensajes de control y mensaje de errores en un .log. De esta manera estos logs estaban recogidos en nuestro local.

8.3.5 Memoria

En la memoria me he encargado de planificar el trabajo y del estudio de las tecnologías que hemos utilizado a lo largo de nuestro Trabajo de Fin de grado.

Con la ayuda de mis compañeros concretamos cuáles eran los principales objetivos de nuestro proyecto y las conclusiones que hemos obtenido.

Además me he encargado de proporcionar propuestas sobre las ampliaciones futuras, elegir cuáles han sido los principales puntos críticos de nuestro trabajo y recoger toda la bibliografía empleada en nuestro proyecto.

Anexo 1: Referencias

Revolución:

<http://www.agentis.es/la-mensajeria-instantanea-revolucion-comunicaciones/>

Tecnología:

<http://www.diariodenavarra.es/noticias/destacados/2017/08/14/la-tecnologia-del-instante-545832-2063.html>

La revolución en las apps:

<http://enpositivo.com/2016/01/las-apps-de-mensajeria-revolucionando-el-mercado/>

Futuro de chatbots:

<http://www.channelpartner.es/corporate/opinion/1099898047402/chatbots-comunicacion-del-futuro-presente.1.html>

<http://www.elplural.com/tech/2017/08/26/los-chatbots-la-gran-esperanza-de-apps-de-mensajeria-instantanea>

Anexo 2: Bibliografía

Telegram:

<https://Telegram.org/>

https://es.wikipedia.org/wiki/Telegram_Messenger

API Telegram:

<https://core.Telegram.org/api>

Telepot API - Autor: Nickoala:

<https://github.com/nickoala/telepot>

WhatsApp:

<https://es.wikipedia.org/wiki/WhatsApp>

<http://www.fotonostra.com/digital/whatsapp.htm>

Facebook Messenger:

https://es.wikipedia.org/wiki/Facebook_Messenger

<https://es-es.messenger.com/>

Python:

<https://es.wikipedia.org/wiki/Python>

<https://www.Python.org/>

<https://www.genbetadev.com/Python/multiprocesamiento-en-Python-threads-a-fondo-introduccion>

http://chuwiki.chuidiang.org/index.php?title=Hilos_en_Python

MySQL:

<https://es.wikipedia.org/wiki/MySQL>

<https://www.mysql.com/>

<https://help.ubuntu.com/lts/serverguide/mysql.html>

PostgreSQL:

<https://www.postgresql.org/>

<https://es.wikipedia.org/wiki/PostgreSQL>

RSS:

<https://es.wikipedia.org/wiki/RSS>

<http://www.RSS.nom.es/>

Google Drive:

https://www.google.com/intl/es_ALL/drive/

https://es.wikipedia.org/wiki/Google_Drive

Sublime text:

https://es.wikipedia.org/wiki/Sublime_Text

Git:

<https://es.wikipedia.org/wiki/Git>

<https://git-scm.com/>

Gedit:

<https://www.guia-ubuntu.com/index.php/Gedit>

Web Scraping

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

https://es.wikipedia.org/wiki/Beautiful_Soup

<http://lxml.de/elementsoup.html>